SILICON UPDATE

Tom Cantrell

A Winter Timer Tale

The weather outside may not exactly be frightful, but that won't stop Tom from settling down with his *TTL Cookbook* and reflecting on what made the '555 so delightful. It's not as old-fashioned as you think.

Ah, the winter season, curling up in front of the fire with an exciting datasheet and the new scope Santa left under the tree. Except here in Silicon Valley, where the weather's so nice I feel like I should be out enjoying it.

Now, before all you snowbirds get envious and pack up to head this way, keep in mind that the nice weather won't be much solace in a few months when folks watch their grass die because they used their meager water ration to wash the car. Anyway, these days "getting out" has devolved into sitting in traffic jams with all the other misguided people.

I'll stick with the datasheet and scope. I don't need the fire, though. Proving my wife wrong, I turn to the dusty bookcases filled with yellowing silicon dreams and aspirations, many subsequently fulfilled but just as many dashed. Admittedly, some of these books haven't been cracked in years (make that decades), but I can't bring myself to chuck them.

I certainly can't take the risk of throwing out a baby like the *TTL Cookbook* by Don Lancaster, with the bath water.[1] Yes dear, you can get away with tossing some obscure items while I'm not looking (Come to think of it, where is that Z8000 manual?), but don't touch my *TTL Cookbook*!

A classic of its time (1974), the *TTL Cookbook* helped tutor an entire generation of neophyte bitheads, providing hours of hands-on entertainment in an era when other diversions were the Brady Bunch and disco.

GOOD VIBRATIONS

Though quaint by today's System-on-Chip standards, the venerable '555, discussed in the *TTL Cookbook*, was a workhorse that found its way into all manner of early digital designs. Remarkably, the same '555 from 25+ years ago still sells today, testimony to the timeless practicality and elegance of the design.

Checking under the hood (see Figure 1), you find a few dozen transistors. The comparator on the right drives the OUTPUT low when the THRESHOLD input reaches two-thirds the supply voltage.

The OUTPUT can also be driven low directly with the RESET input. The low OUTPUT turns on the DISCHARGE transistor. When the TRIG-

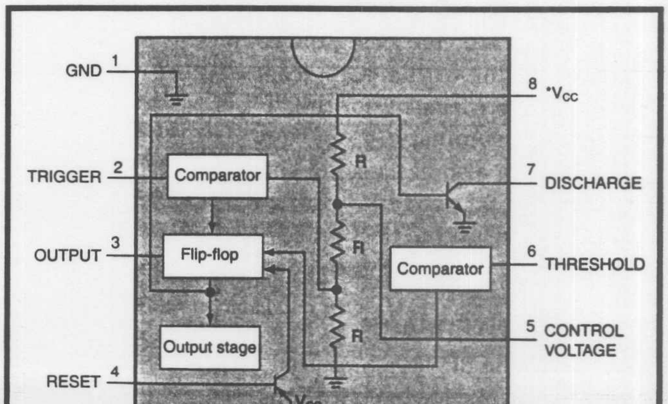

Figure 1—*Timing is everything they say and, though old as the hills, continuing popularity proves that the '555 takes a licking and keeps on ticking.*

GER input goes below one-third the supply voltage, the OUTPUT is driven high, turning off the DISCHARGE transistor. So simple, yet with the addition of a few external resistors and capacitors, so versatile.

Probably the most common application is simply an oscillator, also known in the nomenclature of yore, as an astable multivibrator. By selecting the proper values for two external resistors and a capacitor, you can get anything from close to 1 MHz down to fractions of 1 Hz, with the duty cycle of your choice.

Of course, a '555 oscillator isn't nearly as accurate as a crystal (few percent versus fraction of a percent error) because of tolerances and temperature characteristics of the resistors and capacitors, not to mention variations resulting from a particular PCB layout.

To counter these effects, it's quite common to see a trimpot used in place of one of the resistors to allow the timing to be fine-tuned. In its favor, once dialed in, a '555 is remarkably immune to supply voltage drift, because timing is determined by a voltage ratio rather than by absolute reference.

Wire up the '555 slightly differently and you get a monostable multivibrator (also known as a one shot) that, as the name implies, delivers one cycle of the timed output in response to the trigger. Other circuit tricks and tweaks can turn a '555 into a frequency divider, pulse-width modulator, pulse-position modulator, linear ramp generator, and so on.

Not to say the '555 is perfect. Datasheet specifications and practical limitations (i.e., sticking with standard catalog values) for the external resistor and capacitor cramp the '555's style a little. Also, the range of adjustment (roughly a few microseconds to a few seconds) is relatively wide, but no one would complain if it was wider.

Finally, the ubiquitous trimpot makes for cost, board space, and packaging headaches, not to mention the hassle of manual adjustment.

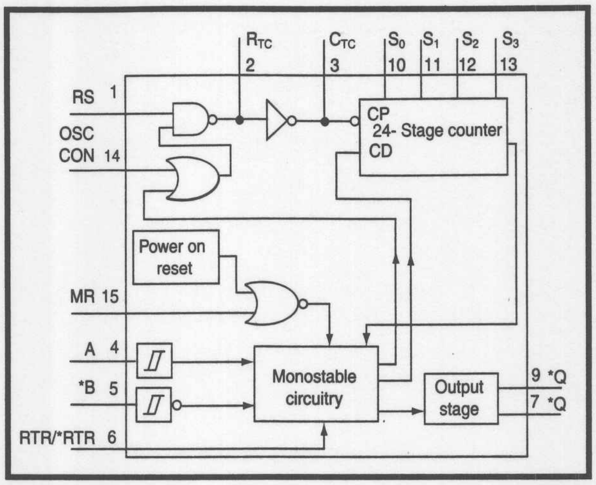

Figure 2—*One of the first of a new breed of digital '555s, the 74HCT5555 freshens the design while remaining true to the spirit of the original '555.*

WELCOME IN THE NEW

Over the years, a number of better '555s have been introduced. For example, wading through the bookcase again, I find the appropriately numbered 74HCT5555 in a 1993 Philips databook (see Figure 2). Old-timers may recall that Signetics, acquired by Philips some years back, was a TTL powerhouse, and Philips has carried that legacy forward.

The '5555 is similar to the '555 in principle, but in practice, it has major improvements. Like other would-be '555 successors, timing is digital, relying on a counter rather than analog circuits.

In the case of the '5555, it's a 24-bit ripple counter, although only sixteen stages (the first and last eight) can be selected. Much like a modern MCU, the oscillator (pins RS, RTC, and CTC) can be driven by an RC, a crystal, or an external input. The oscillator can handle from 1 Hz to 4 MHz using an RC, and from 32 kHz all the way to 20 MHz with a crystal.

Thanks to eight extra pins, the '5555 offers the luxury of both active high and low outputs (Q, *Q) and trigger inputs (A, B), in addition to the reset input (MR). The RTR/*RTR configures the device as retriggerable or not and OSC_CON determines whether the oscillator runs at all times (no startup delay), or only when triggered (low power).

There are some catches. Because accuracy varies depending on the oscillator frequency and divide ratio, it's safe to say that the digital '5555 bests the analog original in terms of both accuracy and range.

I stumbled across another novel pseudo-'555 when I was surfing the Seiko web site to check out the latest on their 7600A embedded 'Net chip (Silicon Update, *Circuit Cellar* 111). I made a quick cutback when I noticed the S-8081B CR Timer (see Figure 3).

The '81B is a simpler variation than the '5555. Although it is digital, the 20-stage divide ratio is fixed, so timing is completely determined by the external resistor and capacitor.

While the '5555 is faster than a '555, the S-8081B is known for being slower. Make that much slower—the datasheet's specified limits for the resistor and capacitor accommodate a timing range from 10 s to 10 h. Of further merit, power consumption is low at only 200 µA, making the chip a good match for battery-driven apps.

DAY-TRIPPER

Enough of this paper chase and the nice weather, I'm itching to wire something up. Fortunately, Santa

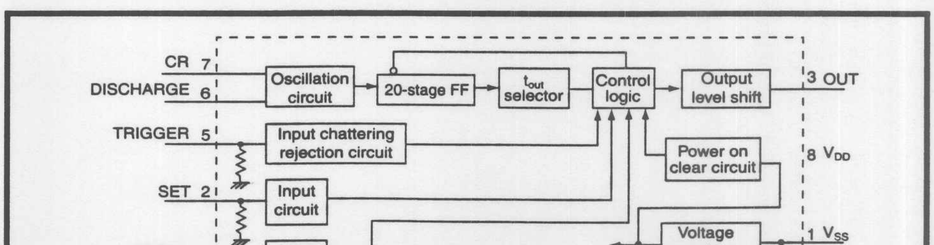

Figure 3—*Long duration timeout (10 s to 10 h) and low-power (200 µA) are the Seiko S8081-B's claim to fame.*

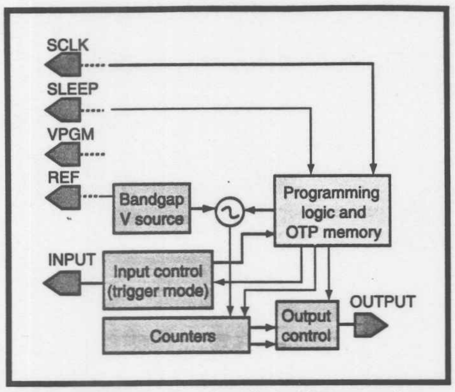

Figure 4—*The latest in a long line of '555 wannabes, the Zilog ZSBI050 incorporates several advances: wider range, clock serial interface, and OTP configuration.*

came through with delivery of a shiny new ZSBI050 from Zilog (see Figure 4). Like the other chips, the basic timing for the OUTPUT is established by an external resistor on the REF pin, but the similarity stops there.

The '050 is unique in three major ways. First, as shown in Figure 5, the complement of dividers is increased to five. Three fixed (two divide-by-1024 and a divide-by-32) dividers can be individually enabled or disabled. The output of the fixed dividers feeds a pair of 10-bit fully programmable divide-by-n counters, the ratio between the two, thus setting the duty cycle.

Second, rather than dedicating pins or compromising the number of selections (as in the '5555), the '050 utilizes a clocked serial interface (SCLK and SDATA) for setting the divide ratios and other operating parameters. Finally, the '050 is kind of an OTP '555, in that it incorporates fuses that allow a configuration to be permanently burned into the chip.

Thirty-five bits of memory define the operating modes and timing. Until programmed, the chip powers up in a default mode (all 0s; i.e., 50/50 oscillator at maximum frequency) using RAM for the configuration. It's an easy matter to pump in 35 bits to configure and calibrate the chip for subsequent OTP programming.

The latter process involves individually shifting in and programming each 1 bit to limit current flow during programming. The programming voltage is 7 V, but because a dedicated VPGM pin can be left open during normal operation, it's easy to accommodate production-line programming.

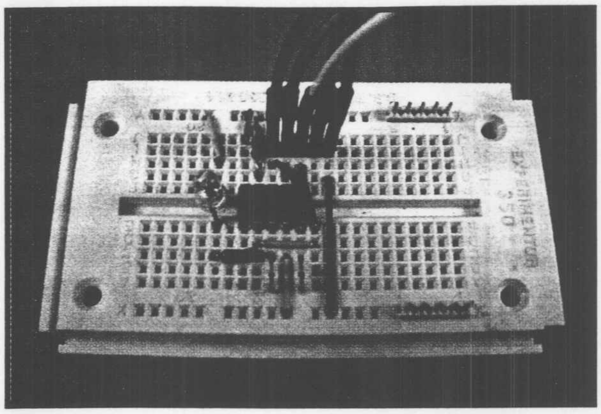

Photo 1—*Eight pins and a simple clock serial interface make the ZSBI050 an easy design-in.*

In one-shot mode, SDATA acts like the trigger input, configurable for rising, falling, or both edges. In oscillator modes, SDATA is a gate input that enables or disables OUTPUT.

Depending on the value of the REF resistor, the oscillator frequency ranges from about 10 kHz to 4 MHz. The OUTPUT frequency is determined by the formula:

$$F_{out} = \frac{F_{osc}}{(2 \times \text{divide-by-}n)+4} \times K$$

where *K* is the product of enabled fixed dividers.

Plug in the numbers and you'll find the possible timing range covers a huge amount of territory, anywhere from a microsecond to more than a month!

TIME-TRACKER

With a rather sparse preliminary datasheet in hand, I had to kind of feel my way with the chip. On the surface, the '050 seemed simple enough with only 8 pins and 35 bits to deal with. However, I've found there's nothing like actually wiring up a chip and getting it working to raise (and answer) the questions the datasheet overlooks.

Verifying the chip is alive and kicking is easy enough. Just hook up 5 V and GND and hang a resistor between REF and GND. On one page of the document, it says the '050 "requires no external components other than one fixed resistor." Yet, in the AC/DC specs, it alludes to the need

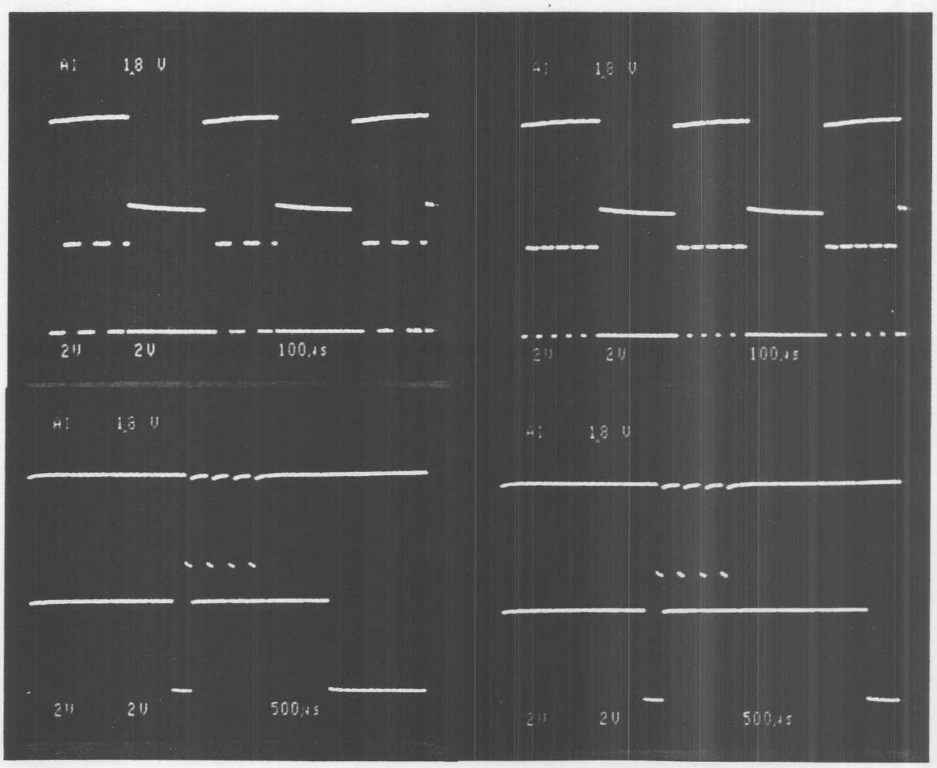

Photo 2—*The ZSBI050 in action. The top trace is SDATA, and the bottom trace Output.* **(a)** *and* **(b)** *show the '050 configured as an oscillator (50/50 and variable duty cycle, respectively) with SDATA used as a gate input.* **(c)** *and* **(d)** *show non-retriggerable and retriggerable one-shot modes with SDATA as the trigger.*

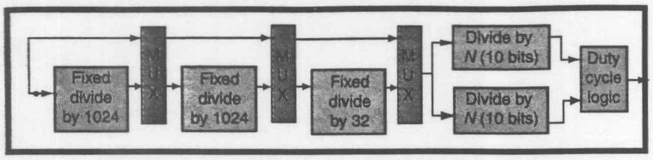

Figure 5—*How slow can you go? With five dividers, the ZSBI050 can hold its breath for a long time.*

for a reference capacitor as well. I decided to believe the less-is-better version and found the '050 seemed to work fine with just a resistor.

Don't forget to connect SDATA to 5 V to enable the OUTPUT, but you can leave SCLK, SLEEP, and VPGM hanging in the breeze. Powerup and, as mentioned earlier, the chip should come up in the default mode of operation as a 50/50 duty cycle oscillator, running at one-quarter the frequency selected with the REF resistor.

I scrounged around and came up with an 82-Ω resistor, which, according to the datasheet, corresponds with about a 200-kHz internal frequency. Sure enough, I saw a nice clean 50-kHz or so square wave on OUTPUT.

Having established that the chip was alive, I proceeded to wire the SCLK, SDATA, and SLEEP lines to a BASIC single-board computer and threw in an LED for kicks (see Photo 1). Just to confirm that SDATA gates the OUTPUT, and SLEEP shuts down the '050.

To configure the '050, I wrote a short program that simply reads 1s and 0s from Data statements and bit-bangs them over to the chip (see Listing 1). The '050 accepts SDATA on the rising edge of SCLK but don't overlook the 10-µs setup and hold time spec. Also, be careful that SDATA only changes state when SCLK is low, lest you inadvertently put the chip into OTP programming mode (invoked by rising edge of SDATA when SCLK is high).

Listing 1—*A simple program is all it takes to configure the ZSBI050. Note the Wait statements that guarantee the required 10-µs setup and hold time.*

```
PROGRAM ZSBI050
INTEGER i,j
CONST zclk = ~WAIT 1:OUT $8000,~
CONST zdata = ~WAIT 1:OUT $8001,~
CONST zslp = ~OUT $8002,~
BEGIN
zclk 0
zdata 0
zslp 0
OUT $8003,$80 /* init PIO */

DATA 0,0,0,0,0 /* reserved */
DATA 0,0 /* mode select */
         /*  0,0 = 50/50 duty cycle oscillator
             0,1 = variable duty cycle oscillator
             1,0 = retriggerable one-shot
             1,1 = non-retriggerable one-shot */
DATA 0,0 /* reserved */
DATA 0,0 /* trigger select */
         /*  0,0 = rising edge
             0,1 = falling edge
             1,x = both edges */
DATA 0   /* divide by 32 */
DATA 0   /* divide by 1024 */
DATA 0   /* divide by 1024 */
DATA 0,0,0,0,0,0,0,1,1,0  /* primary divider */
DATA 0,0,0,0,0,0,0,0,0,0  /* duty cycle */
DATA 0   /* memory protect */

FOR i=1 TO 35
  READ j
  zdata j
  zclk 1
  zclk 0
NEXT i

zdata 1

END
```

I was ready to pull the trigger when I realized I wasn't quite sure who's on first (i.e., the datasheet didn't make the "endian-ness" of the chip real clear). Should I shift the 1st or 35th bit first? Same for multibit fields like the divide ratio—do you send the most or least significant bit first?

"If in doubt, try it out," I say, and it was easy enough to deduce the internal layout by judiciously shifting a single 1 bit. It wasn't long before I was successfully in command of the '050 and able to try out the various features and modes (see Photo 2).

Having explored the '050 operation in RAM mode, I made an admittedly hacker-type attempt to grapple with the programming algorithm and blow the OTP. But, it didn't seem to work. Every time I powered up the part I'd just programmed, it would be the same as one fresh out of the tube.

At the last second, a bit of pondering led me to realize the simple error of my ways. I'd been trying to program a single bit (the divide-by-32 bit). But the question is, how does the '050 know whether to come up in RAM or OTP mode?

It turns out (confirmed by a call to Zilog), that the last bit (Memory Protect) is the key. Its function is described as "preventing further programming" but probably should also include the fact that the OTP is ignored until this bit is set.

Hack the program to blow that 35th bit, and everything works as expected. The divide-by-32 bit had been successfully programmed all along, but the chip was ignoring it.

TIME'S UP

In these days of zillion-transistor chips, you'd think there'd be little need for simple chips like the '555 and its successors, but sometimes a simple tool is best for a simple job. ■

Tom Cantrell has been working on chip, board, and systems design and marketing in Silicon Valley for over a decade. You can reach him by e-mail at tom.cantrell@circuitcellar.com or by telephone at (510) 657-0264.

REFERENCE

[1] D. Lancaster, *TTL Cookbook*, Howard W. Sams & Co., Indianapolis, IN, 1974.

SOURCES

LM555
National Semiconductor
(408) 721-5000
Fax: (408) 739-9803
www.national.com

74HC/T5555
Philips Semiconductors
(408) 991-5207
Fax: (408) 991-3773
www.semiconductor.philips.com

S-8081B
Seiko Instruments ICs
(408) 433-3208
Fax: (408) 433-3214
www.seiko-usa-ecd.com

ZSBI05
Zilog Inc.
(408) 370-8000
Fax: (408) 370-8056
www.zilog.com

CIRCUIT CELLAR

Problem 1—Screw terminal blocks usually come in modular form with two or three terminals per block. The modules can be connected side by side via a tongue and groove on each piece. Often they can be purchased in metric (5.0 mm) or English (5.08 mm) spacings. With such a small difference spacing between pins, either can be used in situations where a low-count terminal module is needed. Why is it important to keep these separated in the parts crib?

Problem 2—The ceramic capacitor size 1206 is the same physical size as the tantalum capacitor size "Y", Johnny is laying out a PCB that uses both. Because both types of capacitors are the same size, can Johnny use the same pad dimentions for both types?

Problem 3—Johnny was perusing the local drugstore's display of batteries. Hanging there on the display were AA, AAA, C, and D size batteries. This got Johnny thinking about the obviously missing A and B size batteries. So, was the pharmacy temporarily out of stock on those items, were these bad designations, or were they discontinued?

Problem 4—On what type of device will you find the notation "REN=x.x", and what does it mean?

What's your EQ?—The answers and 4 additional questions and answers are posted at www.circuitcellar.com.

You may contact the quizmasters at eq@circuitcellar.com.

8 more EQ questions each month in Circuit Cellar Online see pg. 2